BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

US006167471A

United States Patent [19]

Liu et al.

[11] Patent Number: 6,167,471

[45] Date of Patent: Dec. 26, 2000

[54] **METHOD OF AND APPARATUS FOR DISPATCHING A PROCESSING ELEMENT TO A PROGRAM LOCATION BASED ON CHANNEL NUMBER OF RECEIVED DATA**

[75] Inventors: **Jung-Jen Liu**, San Jose; **Bruce A. Fairman**, Woodside, both of Calif.

[73] Assignees: **Sony Corporation**, Tokyo, Japan; **Sony Electronics, Inc.**, Park Ridge, N.J.

[21] Appl. No.: **09/172,994**

[22] Filed: **Oct. 14, 1998**

[51] **Int. Cl.**[7] **G06F 13/10**; G06F 9/30; H04J 3/26; G11C 8/00

[52] **U.S. Cl.** **710/62**; 712/208; 370/464; 455/186.1

[58] **Field of Search** 710/62, 63, 23, 710/100; 711/100; 370/441, 912, 464; 709/300, 229; 712/200, 225, 208; 455/186.1; 375/316; 340/146.2, 825

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,836,722	9/1974	Muller et al. .	
3,906,484	9/1975	Melvin, Jr. et al.	340/347 DD
4,218,756	8/1980	Fraser .	
4,409,656	10/1983	Andersen et al.	364/200
4,493,021	1/1985	Agrawal et al.	364/200
4,641,238	2/1987	Kneib	364/200
4,897,783	1/1990	Nay .	
4,998,245	3/1991	Tanaka et al.	370/85.1
5,008,879	4/1991	Fischer et al.	370/85.2
5,191,418	3/1993	Tran	358/142
5,276,684	1/1994	Pearson	370/94.1
5,325,510	6/1994	Frazier	395/425
5,343,469	8/1994	Ohshima	370/85.1
5,359,713	10/1994	Moran et al.	395/200
5,361,261	11/1994	Edem et al.	370/85.3
5,369,773	11/1994	Hammerstrom	395/800
5,400,340	3/1995	Hillman et al.	370/105.3
5,402,419	3/1995	Osakabe et al.	370/85.1

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

0 428 111 A2	5/1991	European Pat. Off.	G06F 13/28
0 499 394 A1	8/1992	European Pat. Off.	G06F 13/38
0 588 046 A1	3/1994	European Pat. Off.	G06F 13/38
0 696 853 A2	2/1996	European Pat. Off.	H04B 1/20

OTHER PUBLICATIONS

"1394 200 Mb/s PHYsical Layer Transceiver, Product Data Sheet and Application Notes," IBM Microelectronics, Version 1.4, Mar. 14, 1996.

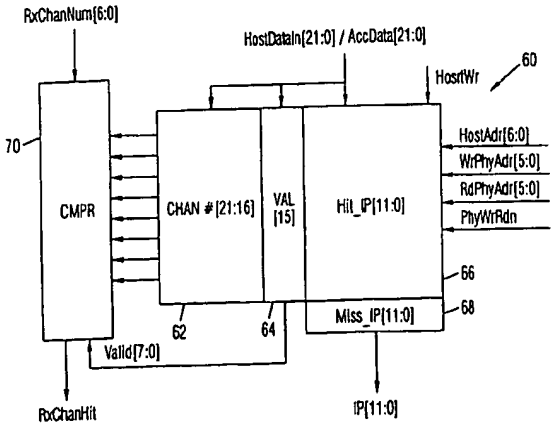
(List continued on next page.)

Primary Examiner—Gopal C. Ray
Attorney, Agent, or Firm—Haverstock & Owens LLP

[57] **ABSTRACT**

An apparatus for dispatching a processing element to a program location based on a channel number of received data includes a channel pointer register having a number of storage locations each with a channel number field, a valid bit field and a corresponding instruction pointer field. When an isochronous channel is allocated for use for reception, the host device programs the channel number and a corresponding instruction pointer value into a storage location. When a storage location is programmed, a valid bit within that storage location is also preferably set. The corresponding instruction pointer value points to a series of instructions which are to be used to process data received on that isochronous channel. When isochronous data is then received, the channel number on which the data is received is compared to the channel numbers within the valid storage locations in the channel pointer register. If one of the channel numbers within a valid storage location matches the channel number of the received data, then the corresponding instruction pointer value is output and the data is processed according to a series of instructions beginning at the location specified by the corresponding instruction pointer value. Otherwise, if the channel number of the received data does not match any of the channel numbers within valid storage locations then a default instruction pointer value is output and the received data is processed according to a series of instructions beginning at the location specified by the default instruction pointer value.

28 Claims, 4 Drawing Sheets



U.S. PATENT DOCUMENTS

5,412,698	5/1995	van Brunt et al.	375/373
5,420,573	5/1995	Tanaka et al.	340/825.24
5,465,402	11/1995	Ono et al. .	
5,493,570	2/1996	Hillman et al.	370/105.3
5,499,344	3/1996	Elnashar et al.	395/250
5,506,846	4/1996	Edem et al.	370/94.2
5,509,126	4/1996	Oprescu et al.	395/307
5,519,701	5/1996	Colmant et al.	370/60.1
5,524,213	6/1996	Dais et al.	395/200.17
5,526,353	6/1996	Henley et al.	370/60.1
5,533,018	7/1996	DeJager et al.	370/60.1
5,535,208	7/1996	Kawakami et al.	370/84
5,537,408	7/1996	Branstad et al.	370/79
5,544,324	8/1996	Edem et al.	395/200.17
5,546,553	8/1996	Robertson et al.	395/405
5,548,587	8/1996	Bailey et al.	370/60.1
5,550,802	8/1996	Worsley et al.	370/13
5,559,796	9/1996	Edem et al.	370/60
5,559,967	9/1996	Oprescu et al.	395/285
5,566,174	10/1996	Sato et al.	370/84
5,586,264	12/1996	Belknap et al.	395/200.08
5,594,732	1/1997	Bell et al.	370/401
5,594,734	1/1997	Worsley et al.	370/395
5,602,853	2/1997	Ben-Michael et al.	370/474
5,603,058	2/1997	Belknap et al.	395/855
5,615,382	3/1997	Gavin et al.	395/800
5,617,419	4/1997	Christensen et al.	370/471
5,619,646	4/1997	Hoch et al.	395/200.01
5,632,016	5/1997	Hoch et al.	395/200.02
5,640,392	6/1997	Hayashi	370/395
5,640,592	6/1997	Rao	395/825
5,646,941	7/1997	Nishimura et al.	370/389
5,647,057	7/1997	Roden et al.	395/275
5,652,584	7/1997	Yoon	341/89
5,655,138	8/1997	Kikinis	395/808
5,659,780	8/1997	Wu	395/800.19
5,664,124	9/1997	Katz et al.	395/309
5,668,948	9/1997	Belknap et al.	395/200.61
5,684,954	11/1997	Kaiserwerth et al.	395/200.2
5,687,316	11/1997	Graziano et al.	395/200.2
5,689,727	11/1997	Bonke et al.	395/840
5,692,211	11/1997	Gulick et al.	395/800
5,696,924	12/1997	Robertson et al.	395/412
5,704,052	12/1997	Wu et al.	395/380
5,706,439	1/1998	Parker	395/200.17
5,708,779	1/1998	Graziano et al.	395/200.8
5,710,773	1/1998	Shiga	370/512
5,752,076	5/1998	Munson	395/825
5,758,075	5/1998	Graziano et al.	395/200.8
5,761,457	6/1998	Gulick	395/308
5,774,683	6/1998	Gulick	395/309
5,781,599	7/1998	Shiga	375/376
5,793,953	8/1998	Yeung et al.	395/200.8
5,815,678	9/1998	Hoffman et al.	395/309
5,828,416	10/1998	Ryan .	
5,828,903	10/1998	Sethuram et al.	395/817
5,835,726	11/1998	Shwed et al.	395/200.59
5,835,793	11/1998	Li et al.	395/898
5,848,253	12/1998	Walsh et al.	395/309
5,884,103	3/1999	Terho et al.	395/892
5,938,752	8/1999	Leung et al.	710/126
5,946,298	8/1999	Okuyama	370/232
5,970,236	10/1999	Galloway et al.	395/500

OTHER PUBLICATIONS

"TSB21LV03 IEEE 1394—1995 Triple Cable Transceiver/Arbiter," Texas Instruments, Product Preview, Revision 0.99, Mar. 19, 1996.

"The IEEE—1394 High Speed Serial Bus," R.H.J. Bloks, Philips Journal of Research, vol. 50, No. 1/2, 1996.

"PC 4 intern Systemprogrammierung," Data Becker, 1994.

"IEEE 1394, The Cable Connection to Complete the Digital Revolution," Daniel Moore.

"The Parallel Protocol Engine" Matthias Kaiserswerth, IEEE/ACM Transactions on Networking, Dec. 1993, New York, pp. 650–663.

"The Programmable Protocol VLSI Engine (PROVE)" A.S. Krishnakumar, W.C. Fischer, and Krishan Sabnani, IEEE Transactions on Communications, Aug. 1994, New York, pp. 2630–2642.

"A Bus on a Die—The Serial Bus Alternative" Michael Teener, ComCon92, Feb. 24–28, 1992, pp. 316–321.

"Local Area Network Protocol for Autonomous Control of Attached Devices" Software Patent Institute, 1995, 1996 Publication Date Jul., 1990.

"Architecture for High Performance Transparent Bridges" Software Patent Institute, 1995, 1996. Publication Date Jul., 1992.

"Access to High–Speed LAN via Wireless Media" Software Patent Institute, 1995, 1996. Publication Date Apr., 1993.

"Asynchronous Transfer Mode" Julia L. Heeter, Dec. 12, 1995.

"The SerialSoft IEEE 1394 Developer Tool" Skipstone.

"Data link driver program design for the IBM token ring network PC adapter" Gee–Swee Poo and Wilson Ang, Computer Communications, 1989, London, Great Britain, pp. 266–272.

"Fiber Channel (FCS)/ATM interworking: A design solution" A. Anzaloni, M. De Sanctis, F. Avaltroni, G. Rulli, L. Proietti and G. Lombardi, Ericsson Fatme R&D Division, Nov. 1993, pp. 1127–1133.

"Data Exchange Adapter for Micro Channel/370" Software Patent Institute, 1995, 1996. Publication Date Oct., 1991.

IEEE, "1394 Standard for a High Performance Serial Bus," 1995, USA.

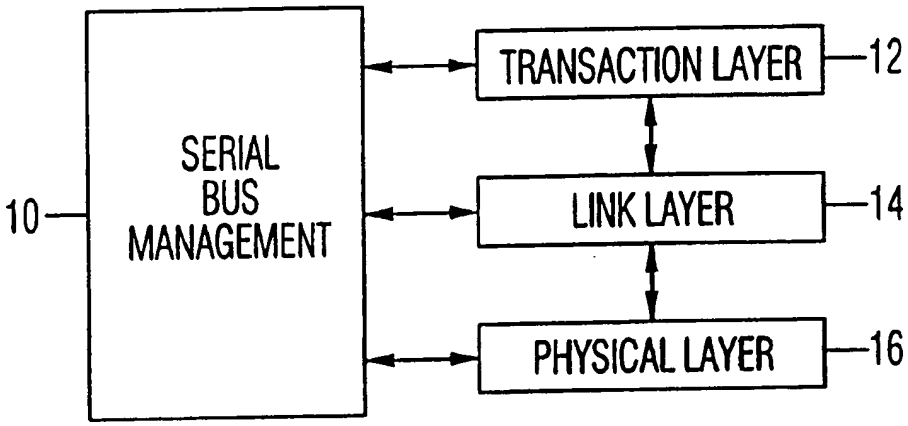


Fig. 1
(Prior Art)

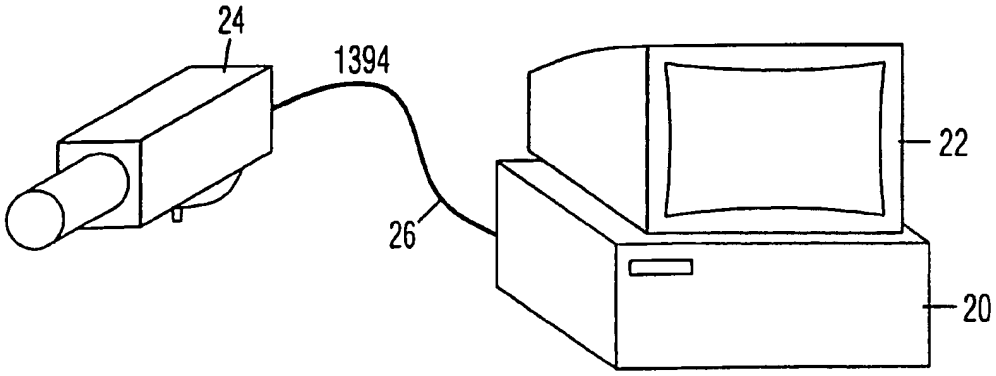


Fig. 2

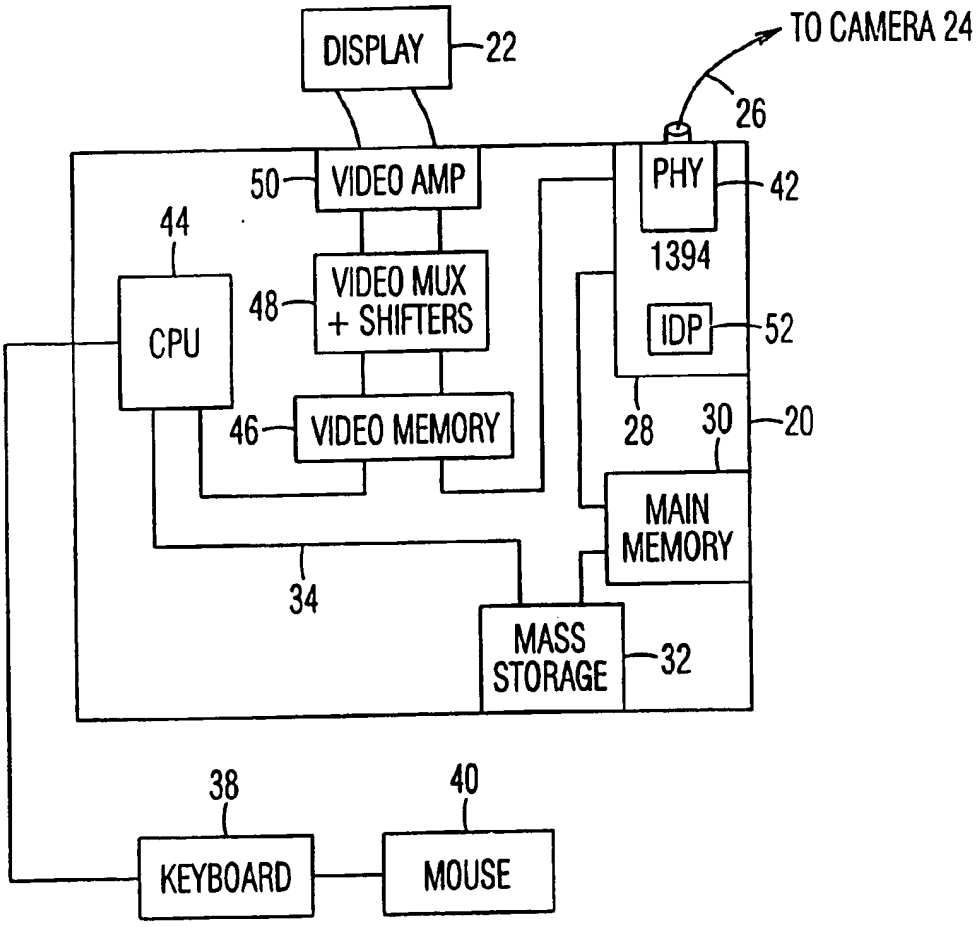

DISPLAY
22
TO CAMERA 24
26
50
VIDEO AMP
PHY
42
1394
44
48
VIDEO MUX
+ SHIFTERS
CPU
IDP
52
46
VIDEO MEMORY
28
30
20
MAIN
MEMORY
34
MASS
STORAGE
32
38
40
KEYBOARD
MOUSE

Fig. 3

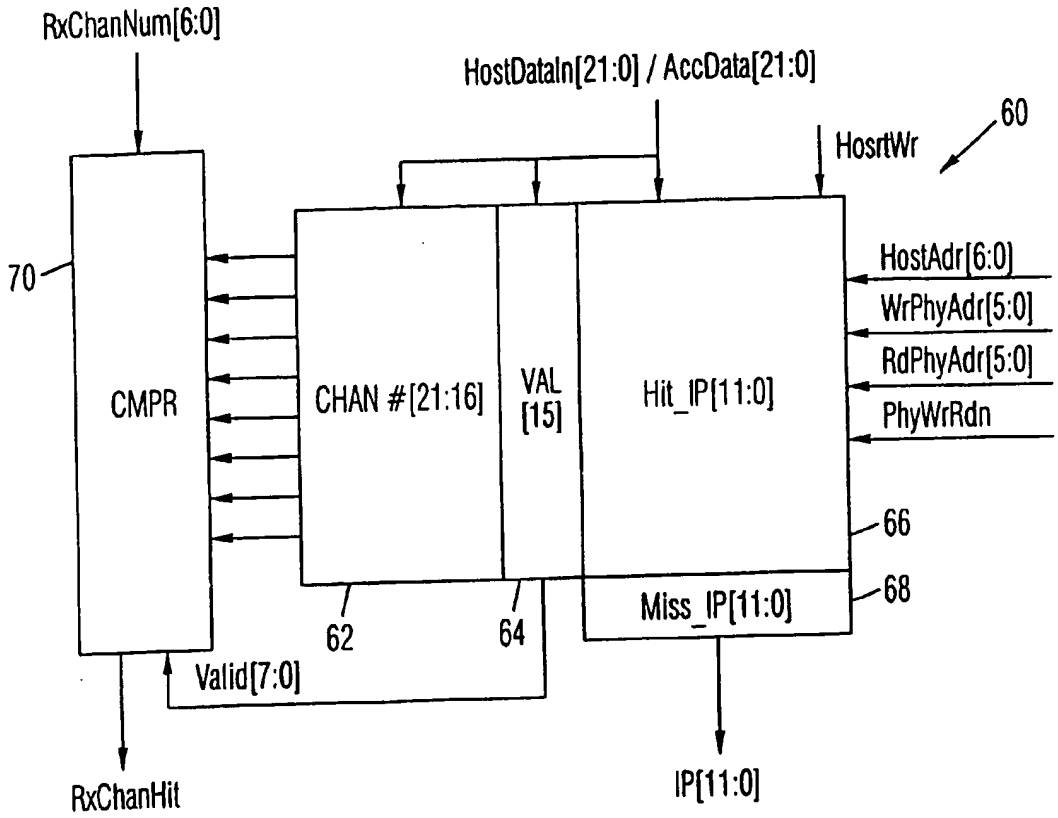

RxChanNum[6:0]
HostDataIn[21:0] / AccData[21:0]
HosrtWr
60
70
CMPR
CHAN #[21:16]
VAL [15]
Hit_IP[11:0]
HostAdr[6:0]
WrPhyAdr[5:0]
RdPhyAdr[5:0]
PhyWrRdn
66
Miss_IP[11:0]
68
62
64
Valid[7:0]
RxChanHit
IP[11:0]

Fig. 4

METHOD OF AND APPARATUS FOR DISPATCHING A PROCESSING ELEMENT TO A PROGRAM LOCATION BASED ON CHANNEL NUMBER OF RECEIVED DATA

FIELD OF THE INVENTION

The present invention relates to the field of managing the reception of data received by a device. More particularly, the present invention relates to the field of managing the reception of data received on a channel by a device.

BACKGROUND OF THE INVENTION

The IEEE standard, "IEEE 1394 Standard For A High Performance Serial Bus," Draft ratified in 1995, is an international standard for implementing an inexpensive high-speed serial bus architecture which supports both asynchronous and isochronous format data transfers. Isochronous data transfers are real-time transfers which take place such that the time intervals between significant instances have the same duration at both the transmitting and receiving applications. Each packet of data transferred isochronously is transferred in its own time period. The IEEE 1394-1995 standard bus architecture provides up to sixty-four (64) channels for isochronous data transfer between applications. A six bit channel number is broadcast with the data to ensure reception by the appropriate application. This allows multiple applications to simultaneously transmit isochronous data across the bus structure. Asynchronous transfers are traditional data transfer operations which take place as soon as possible and transfer an amount of data from a source to a destination.

The IEEE 1394-1995 standard provides a high-speed serial bus for interconnecting digital devices thereby providing a universal I/O connection. The IEEE 1394-1995 standard defines a digital interface for the applications thereby eliminating the need for an application to convert digital data to analog data before it is transmitted across the bus. Correspondingly, a receiving application will receive digital data from the bus, not analog data, and will therefore not be required to convert analog data to digital data. The cable required by the IEEE 1394-1995 standard is very thin in size compared to other bulkier cables used to connect such devices. Devices can be added and removed from an IEEE 1394-1995 bus while the bus is active. If a device is so added or removed the bus will then automatically reconfigure itself for transmitting data between the then existing nodes. A node is considered a logical entity with a unique identification number on the bus structure. Each node provides an identification ROM, a standardized set of control registers and its own address space.

The IEEE 1394-1995 standard defines a protocol as illustrated in FIG. 1. This protocol includes a serial bus management block **10** coupled to a transaction layer **12**, a link layer **14** and a physical layer **16**. The physical layer **16** provides the electrical and mechanical connection between a device or application and the IEEE 1394-1995 cable. The physical layer **16** also provides arbitration to ensure that all devices coupled to the IEEE 1394-1995 bus have access to the bus as well as actual data transmission and reception. The link layer **14** provides data packet delivery service for both asynchronous and isochronous data packet transport. This supports both asynchronous data transport, using an acknowledgement protocol, and isochronous data transport, providing real-time guaranteed bandwidth protocol for just-in-time data delivery. The transaction layer **12** supports the commands necessary to complete asynchronous data transfers, including read, write and lock. The serial bus management block **10** contains an isochronous resource manager for managing isochronous data transfers. The serial bus management block **10** also provides overall configuration control of the serial bus in the form of optimizing arbitration timing, guarantee of adequate electrical power for all devices on the bus, assignment of the cycle master, assignment of isochronous channel and bandwidth resources and basic notification of errors.

As discussed above, an IEEE 1394-1995 device includes the capability to transmit and receive isochronous data over multiple channels. The IEEE 1394-1995 standard provides for up to sixty-four different isochronous channels to be used within an IEEE 1394-1995 network of devices. However, in current implementations, certain IEEE 1394-1995 devices are being built with the capability to only transmit and receive isochronous data over a subset of less than sixty-four channels. When receiving data on an isochronous channel, that data must be processed by the receiving device. This processing includes any or all of displaying, manipulating, forwarding and storing. Often, data received on different isochronous channels is processed differently, depending on the type of device from which the data is received, the type of data that is received and the desired use of the data. If data received on an isochronous channel is not received and processed efficiently, errors in the display or use of the data can result.

SUMMARY OF THE INVENTION

An apparatus for dispatching a processing element to a program location based on a channel number of received data includes a channel pointer register having a number of storage locations each with a channel number field, a valid bit field and a corresponding instruction pointer field. When an isochronous channel is allocated for use for reception, the host device programs the channel number and a corresponding instruction pointer value into a storage location. When a storage location is programmed, a valid bit within that storage location is also preferably set. The corresponding instruction pointer value points to a series of instructions which are to be used to process data received on that isochronous channel. When isochronous data is then received, the channel number on which the data is received is compared to the channel numbers within the valid storage locations in the channel pointer register. If one of the channel numbers within a valid storage location matches the channel number of the received data, then the corresponding instruction pointer value is output and the data is processed according to a series of instructions beginning at the location specified by the corresponding instruction pointer value. Otherwise, if the channel number of the received data does not match any of the channel numbers within valid storage locations then a default instruction pointer value is output and the received data is processed according to a series of instructions beginning at the location specified by the default instruction pointer value.

In one aspect of the invention, a method of processing received data comprises the steps of receiving data on a received channel number, comparing the received channel number to stored channel numbers within a plurality of memory locations, each of the plurality of memory locations including a corresponding address value specifying a starting address for a series of instructions for processing data received on a corresponding stored channel number, providing the corresponding address value corresponding to the stored channel number matching the received channel number as an output address value if one of the stored channel

numbers matches the received channel number and providing a default address value as the output address value if none of the stored channel numbers matches the received channel number. The method further includes the steps of allocating an allocated channel number for receiving data and programming the allocated channel number and a corresponding allocated address value into one of the memory locations to form the stored channel number and the corresponding address value for the memory location. The default address value specifies a default starting address for a series of default instructions for processing data received on the received channel number. The memory locations are locations within a register. The method further includes the step of programming a valid bit within a programmed memory location. The data is preferably isochronous data.

In another aspect of the invention, an apparatus for processing received data comprises a plurality of storage locations each including a channel number field to store a stored channel number and an instruction pointer field to store a stored address value, a comparing circuit coupled to the plurality of storage locations and configured to receive a received channel number on which data is received, wherein the comparing circuit compares the stored channel numbers to the received channel number to determine if any of the stored channel numbers match the received channel number and an output circuit coupled to the comparing circuit and to the plurality of storage locations to provide the stored address value within the storage location having the stored channel number matching the received channel number as an output address value. The apparatus further comprises a default storage location coupled to the output circuit to store a default address value and provide the default address value as the output address value if none of the stored channel numbers match the received channel number. The storage locations further each include a valid bit. The storage locations are programmable. The apparatus further comprises a host device coupled to the plurality of storage locations to program the stored channel numbers and the stored address values. The host device sets the valid bit within the storage location when the storage location is programmed. The apparatus further comprises a processing device coupled to the output circuit to receive the output address value, wherein the output address value specifies a beginning location for a series of instructions to be used to process the data received on the received channel number. The data is preferably isochronous data.

In still another aspect of the invention, an apparatus for processing received data comprises means for storing including a plurality of storage locations each having a channel number field to store a stored channel number and an instruction pointer field to store a stored address value, means for comparing coupled to the means for storing and configured for receiving a received channel number on which data is received, wherein the means for comparing compares the stored channel numbers to the received channel number to determine if any of the stored channel numbers match the received channel number and means for providing coupled to the means for comparing and to the means for storing for providing the stored address value within the storage location having the stored channel number matching the received channel number as an output address value. The apparatus further comprises a default storage location coupled to the means for providing for storing a default address value and providing the default address value as the output address value if none of the stored channel numbers match the received channel number. The storage locations further each include a valid bit. The storage locations are programmable. The apparatus further comprises a host device coupled to the means for storing for programming the stored channel numbers and the stored address values. The host device sets the valid bit within the storage location when the storage location is programmed. The apparatus further comprises a processing device coupled to the means for providing to receive the output address value, wherein the output address value specifies a beginning location for a series of instructions to be used to process the data received on the received channel number. The means for storing is a register. The data is preferably isochronous data.

In another aspect of the invention, a receiving device for receiving data from one or more remote devices comprises an interface circuit configured to receive data on one or more received channel numbers, a plurality of storage locations each including a channel number field to store a stored channel number, an instruction pointer field to store a stored address value and a valid bit having a first state and a second state, wherein when the valid bit for a storage location is in the first state the storage location is valid, a comparing circuit coupled to the interface circuit and to the plurality of storage locations to receive the received channel number corresponding to received data and compare the stored channel numbers to the received channel number to determine if any of the stored channel numbers match the received channel number, a default storage location for storing a default address value and an output circuit coupled to the comparing circuit, to the plurality of storage locations and to the default storage location to provide the stored address value within the storage location having the stored channel number matching the received channel number as an output address value, if one of the stored channel numbers within valid storage locations matches the received channel number, and to provide the default address value as the output address value if none of the stored channel numbers within the valid storage locations match the received channel number. The comparing circuit only compares stored channel numbers within the valid storage locations to the received channel number. The storage locations are programmable. The host device sets the valid bit within the storage location when the storage location is programmed. The receiving device further comprises a processing device coupled to the output circuit to receive the output address value, wherein the output address value specifies a beginning location for a series of instructions to be used to process the data received on the received channel number.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** illustrates a protocol of the IEEE 1394-1995 standard.

FIG. **2** illustrates an exemplary IEEE 1394-1995 serial bus network including a computer system and a video camera.

FIG. **3** illustrates a block diagram of the internal components of the computer system **20**.

FIG. **4** illustrates a channel pointer register according to the preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

An apparatus for dispatching a processing element to a program location based on a channel number of received data includes a channel pointer register. The channel pointer register includes a number of storage locations each having a channel number field, a valid bit field and a corresponding

instruction pointer field. Each of the storage locations is programmed by a host device when the host device allocates an isochronous channel and expects to begin receiving data on the isochronous channel. When a storage location within the channel pointer register is programmed, the channel number on which the host device expects to receive isochronous data is written into the channel number field and a corresponding instruction pointer value is written into the instruction pointer field. The corresponding instruction pointer value is a program location preferably recognized by an isochronous data pipe (IDP), which includes program instructions for processing of data received on this isochronous channel. When a storage location is programmed with a channel number and corresponding instruction pointer value, the valid bit within the valid bit field is set to a logical high voltage level. Only storage locations with a valid bit having a value equal to a logical low voltage level can be programmed. When the data within a storage location is no longer valid or an isochronous channel has been reallocated, the valid bit within that storage location is reset to a logical low voltage level, signalling that the data within that storage location is no longer valid.

When isochronous data is received by the host device, the host device determines on which channel the isochronous data has been received. This channel number is then compared to the channel numbers within the valid storage locations in the channel pointer register. If one of the channel numbers within a valid storage location matches the current receive channel number, then the channel pointer register outputs the instruction pointer value within the matching storage location. This instruction pointer value points to a program location. The program instructions beginning at this program location are then used by the host device to process the data received on the current receive channel number. Preferably, the host device utilizes an isochronous data pipe to process the received isochronous data, as taught within U.S. patent application Ser. No. 08/612,322, filed on Mar. 7, 1996 and entitled "ISOCHRONOUS DATA PIPE FOR MANAGING AND MANIPULATING A HIGH-SPEED STREAM OF ISOCHRONOUS DATA FLOWING BETWEEN AN APPLICATION AND A BUS STRUCTURE," which is hereby incorporated by reference. Alternatively, any other appropriate processing device can be used to process the incoming isochronous data.

If none of the channel numbers within the valid storage locations in the channel pointer register match the current receive channel number, then the channel pointer register outputs an instruction pointer default value. This instruction pointer default value points to a default program location. The default program instructions beginning at this program location are then used by the IDP or other appropriate processing device to process the data received on the current receive channel number.

An exemplary IEEE 1394-1995 serial bus network implementing the present invention and including a computer system and a video camera is illustrated in FIG. **2**. The computer system **20** includes an associated display **22** and is coupled to the video camera **24** by the IEEE 1394-1995 serial bus cable **26**. Video data and associated data are sent between the video camera **24** and the computer system **20** over the IEEE 1394-1995 serial bus cable **26**.

A block diagram of the internal components of the computer system **20** is illustrated in FIG. **3**. The computer system **20** includes a central processor unit (CPU) **44**, a main memory **30**, a video memory **46**, a mass storage device **32** and an IEEE 1394-1995 interface circuit **28**, all coupled together by a conventional bidirectional system bus **34**. The interface circuit **28** includes the physical interface circuit **42** for sending and receiving communications on the IEEE 1394-1995 serial bus and the isochronous data pipe **52** which is used to process streams of isochronous data received and transmitted through the physical interface circuit **42**. The physical interface circuit **42** is coupled to the camera **24** over the IEEE 1394-1995 serial bus cable **26**. The system bus **34** contains an address bus for addressing any portion of the memory **30** and **46**. The system bus **34** also includes a data bus for transferring data between and among the CPU **44**, the main memory **30**, the video memory **46**, the mass storage device **32** and the interface circuit **28**.

The computer system **20** is also coupled to a number of peripheral input and output devices including the keyboard **38**, the mouse **40** and the associated display **22**. The keyboard **38** is coupled to the CPU **44** for allowing a user to input data and control commands into the computer system **20**. A conventional mouse **40** is coupled to the keyboard **38** for manipulating graphic images on the display **22** as a cursor control device. As is well known in the art, the mouse **40** can alternatively be coupled directly to the computer **20** through a serial port.

A port of the video memory **46** is coupled to a video multiplex and shifter circuit **48**, which in turn is coupled to a video amplifier **50**. The video amplifier **50** drives the display **22**. The video multiplex and shifter circuitry **48** and the video amplifier **50** convert pixel data stored in the video memory **46** to raster signals suitable for use by the display **22**.

The channel pointer register of the present invention is illustrated in FIG. **4**. Preferably, the channel pointer register resides within the main memory **30** of the host device. Alternatively, the channel pointer register is provided within a dedicated register. The channel pointer register **60** is preferably a twenty-two bit register having eight storage locations each including a channel number field **62**, a valid bit field **64** and a corresponding instruction pointer field **66**. A corresponding instruction pointer value is stored within the corresponding instruction pointer field **66** in bits **0** through **11** of the storage location. The valid bit is stored within the valid bit field **64** in bit **15** of the storage location. The channel number is stored within the channel number field **62** in bits **16** through **21** of the storage location. Bits **12–14** of the storage locations are preferably reserved for future use. The channel pointer register **60** also includes a default instruction pointer value storage location **68** in which a default instruction pointer value is stored for use when the channel number of received data does not match any of the valid channel numbers stored within the channel pointer register **60**.

A host data input signal HostDataIn is coupled to the channel pointer register **60** in order to program the storage locations within the channel pointer register **60**. The host data input signal HostDataIn is provided by the host device to program the storage locations with the appropriate data for isochronous channels on which the host device is receiving data. A host write strobe input HostWr is coupled to the channel pointer register **60** for strobing data into the channel pointer register **60**. A host address input signal HostAdr is provided by the host and specifies at which storage location within the channel pointer register **60** the data on the host data input signal HostDataIn is to be written. The channel pointer register **60** provides a twelve-bit instruction pointer output signal IP, depending on the channel number of the received data.

The storage locations within the channel pointer register **60** are also read from and written to by an arithmetic logic

unit (ALU) within the IDP **52** or other processing element within the host device. An accumulator data input signal AccData is also coupled to the channel pointer register **60** to program storage locations within the channel pointer register **60**. A write physical address input signal WrPhyAdr, a read physical address input signal RdPhyAdr and a register physical write/read strobe input signal PhyWrRdn are coupled to the channel pointer register **60** and used by the ALU to read from and write data to the storage locations within the channel pointer register **60**.

Each storage location within the channel pointer register **60** is coupled to a channel number comparator **70** for comparing the channel numbers within the channel pointer register **60** with the channel number on which the host device is currently receiving data. The valid bit values for each storage location within the channel pointer register **60** are also provided to the channel number comparator **70** on the signal lines Valid. Using the valid bit values provided on the signal lines Valid, the channel number comparator **70** can readily determine which storage locations within the channel pointer register **60** include a valid channel number and corresponding instruction pointer. A six-bit receive channel number input signal RxChanNum is coupled to the channel number comparator **70**. The receive channel number input signal RxChanNum is provided from the IDP **52** or other processing element within the host device and specifies the channel number on which data is currently being received. A receive channel hit output signal RxChanHit is provided from the channel number comparator **70**. The receive channel hit output signal RxChanHit is activated and raised to a logical high voltage level by the comparator **70** when the isochronous channel on which data is currently being received matches a channel number within one of the valid storage locations in the channel pointer register **60**. The receive channel hit output signal RxChanHit is deactivated and pulled to a logical low voltage level by the comparator **70** when the isochronous channel on which data is currently being received does not match any of the channel numbers within the valid storage locations in the channel pointer register **60**.

When programming a storage location within the channel pointer register **60**, the host device puts the data to be written into the storage location on the host data input signal HostDataIn. The host device also provides the address of the storage location to be written to on the host address input signal HostAdr. When the address of the storage location within the channel pointer register is on the host address input signal HostAdr and the data to be written into the storage location is on the host data input signal HostDataIn, the host device then activates the host write strobe signal HostWr. When the host write strobe signal HostWr is activated, the data on the host data input signal HostDataIn is written into the storage location within the channel pointer register **60** specified by the host address input signal HostAdr. In this manner, the storage locations within the channel pointer register **60** are programmed by the host device.

The data written into the storage location includes the channel number value which is written into the channel number field **62** and the corresponding instruction pointer value which is written into the instruction pointer field **66**. When a storage location is programmed with a channel number value and a corresponding instruction pointer value, the valid bit is also set signalling that the data within that storage location is valid for the channel number specified. During programming, different channel numbers are written into the channel number field **62** of different storage locations.

When receiving data, the channel number on which the data is received is provided to the comparator **70** on the receive channel number input signal RxChanNum. The comparator **70** then compares the channel number received on the receive channel number input signal RxChanNum to the channel numbers within the valid storage locations in the channel pointer register **60**. If one of the channel numbers within a storage location having a valid bit set to a logical high voltage level matches the channel number on the receive channel number input signal RxChanNum, then the instruction pointer value stored within that storage location is output on the instruction pointer output signal IP. Also, if one of the channel numbers within a valid storage location matches the channel number input on the receive channel input signal RxChanNum, then the comparator raises the receive channel hit output signal RxChanHit to a logical high voltage level, signalling that the received channel number was matched. Otherwise, if none of the channel numbers within valid storage locations match the channel number input on the receive channel input signal RxChanNum, then the default instruction pointer value stored within the default instruction pointer value storage location **68** is output on the instruction pointer output signal IP and the comparator **70** pulls the receive channel hit output signal RxChanHit to a logical low voltage level, signalling that the received channel number was not matched.

The value output on the instruction pointer output signal IP is preferably provided to the IDP **52**. The IDP **52** runs a set of programmed instructions to process the incoming isochronous data, beginning at the address specified by the instruction pointer output signal IP. Alternatively, any other processing device can be used to process the data and execute the series of instructions starting at the address specified by the instruction pointer output signal IP.

When an isochronous channel is allocated between the host device and a remote device on the IEEE 1394-1995 serial bus network, the host device programs the channel number and the corresponding instruction pointer value into a storage location within the channel pointer register **60** for the allocated channel. The corresponding instruction pointer value points to a series of instructions which are to be performed on data received on that channel. The host device also sets the valid bit for that storage location. When data is then received on the allocated channel, the IDP **52** or other processing element within the host device provides the channel number to the comparator **70**. The comparator **70** then compares the channel number to the channel numbers within valid storage locations in the channel pointer register **60**. If the channel number matches a channel number within a valid storage location, then the instruction pointer value within that storage location is output on the instruction pointer output signal IP. The IDP within the host device then processes the isochronous data received on that channel according to the series of instructions at the location specified by the instruction pointer output signal IP.

Otherwise, if the channel number does not match a channel number within a valid storage location, then the default instruction pointer value within the default storage location **68** is output on the instruction pointer output signal IP. Preferably, the IDP within the host device then processes the isochronous data received on that channel according to the series of instructions starting at the location specified by the default instruction pointer value.

Using the method and apparatus of the present invention, a host device can efficiently process incoming isochronous data. The host device programs channel numbers and corresponding instruction pointer values into the channel

pointer register **60**. When data is then received on an isochronous channel, the channel number on which that data is received is input to the comparator **70** and compared with the channel numbers in valid storage locations within the channel pointer register **60**. If one of the channel numbers within a valid storage location matches the channel number of the received data, then the corresponding instruction pointer value is output and the data is processed according to a series of instructions beginning at the location pointed to by the corresponding instruction pointer value. If the channel number of the received data does not match any of the channel numbers within valid storage locations in the channel pointer register then the default instruction pointer value is output and the data is processed according to a series of instructions beginning at the location pointed to by the default instruction pointer value.

The present invention has been described in terms of specific embodiments incorporating details to facilitate the understanding of principles of construction and operation of the invention. Such reference herein to specific embodiments and details thereof is not intended to limit the scope of the claims appended hereto. It will be apparent to those skilled in the art that modifications may be made in the embodiment chosen for illustration without departing from the spirit and scope of the invention. Specifically, it will be apparent to those skilled in the art that while the preferred embodiment of the present invention is used with an IEEE 1394-1995 serial bus structure, the present invention could also be implemented on any other appropriate bus structures.

We claim:

1. A method of processing received data comprising the steps of:

a. receiving data on a received channel number;

b. comparing the received channel number to stored channel numbers within a plurality of memory locations, each of the plurality of memory locations including a corresponding address value specifying a starting address for a series of instructions for processing data received on a corresponding stored channel number;

c. providing the corresponding address value corresponding to the stored channel number matching the received channel number as an output address value if one of the stored channel numbers matches the received channel number; and

d. providing a default address value as the output address value if none of the stored channel numbers matches the received channel number.

2. The method as claimed in claim **1** further comprising the steps of:

a. allocating an allocated channel number for receiving data; and

b. programming the allocated channel number and a corresponding allocated address value into one of the memory locations to form the stored channel number and the corresponding address value for the memory location.

3. The method as claimed in claim **2** wherein the default address value specifies a default starting address for a series of default instructions for processing data received on the received channel number.

4. The method as claimed in claim **3** wherein the memory locations are locations within a register.

5. The method as claimed in claim **4** further comprising the step of programming a valid bit within a programmed memory location.

6. The method as claimed in claim **5** wherein the data is isochronous data.

7. An apparatus for processing received data comprising:

a. a plurality of storage locations each including a channel number field to store a stored channel number and an instruction pointer field to store a stored address value;

b. a comparing circuit coupled to the plurality of storage locations and configured to receive a received channel number on which data is received, wherein the comparing circuit compares the stored channel numbers to the received channel number to determine if any of the stored channel numbers match the received channel number; and

c. an output circuit coupled to the comparing circuit and to the plurality of storage locations to provide the stored address value within the storage location having the stored channel number matching the received channel number as an output address value.

8. The apparatus as claimed in claim **7** further comprising a default storage location coupled to the output circuit to store a default address value and provide the default address value as the output address value if none of the stored channel numbers match the received channel number.

9. The apparatus as claimed in claim **8** wherein the storage locations further each include a valid bit.

10. The apparatus as claimed in claim **9** wherein the storage locations are programmable.

11. The apparatus as claimed in claim **10** further comprising a host device coupled to the plurality of storage locations to program the stored channel numbers and the stored address values.

12. The apparatus as claimed in claim **11** wherein the host device sets the valid bit within the storage location when the storage location is programmed.

13. The apparatus as claimed in claim **12** further comprising a processing device coupled to the output circuit to receive the output address value, wherein the output address value specifies a beginning location for a series of instructions to be used to process the data received on the received channel number.

14. The apparatus as claimed in claim **13** wherein the data is isochronous data.

15. An apparatus for processing received data comprising:

a. means for storing including a plurality of storage locations each having a channel number field to store a stored channel number and an instruction pointer field to store a stored address value;

b. means for comparing coupled to the means for storing and configured for receiving a received channel number on which data is received, wherein the means for comparing compares the stored channel numbers to the received channel number to determine if any of the stored channel numbers match the received channel number; and

c. means for providing coupled to the means for comparing and to the means for storing for providing the stored address value within the storage location having the stored channel number matching the received channel number as an output address value.

16. The apparatus as claimed in claim **15** further comprising a default storage location coupled to the means for providing for storing a default address value and providing the default address value as the output address value if none of the stored channel numbers match the received channel number.

17. The apparatus as claimed in claim **16** wherein the storage locations further each include a valid bit.

18. The apparatus as claimed in claim 17 wherein the storage locations are programmable.

19. The apparatus as claimed in claim 18 further comprising a host device coupled to the means for storing for programming the stored channel numbers and the stored address values.

20. The apparatus as claimed in claim 19 wherein the host device sets the valid bit within the storage location when the storage location is programmed.

21. The apparatus as claimed in claim 20 further comprising a processing device coupled to the means for providing to receive the output address value, wherein the output address value specifies a beginning location for a series of instructions to be used to process the data received on the received channel number.

22. The apparatus as claimed in claim 21 wherein the means for storing is a register.

23. The apparatus as claimed in claim 22 wherein the data is isochronous data.

24. A receiving device for receiving data from one or more remote devices comprising:

a. an interface circuit configured to receive data on one or more received channel numbers;

b. a plurality of storage locations each including a channel number field to store a stored channel number, an instruction pointer field to store a stored address value and a valid bit having a first state and a second state, wherein when the valid bit for a storage location is in the first state the storage location is valid;

c. a comparing circuit coupled to the interface circuit and to the plurality of storage locations to receive the received channel number corresponding to received data and compare the stored channel numbers to the received channel number to determine if any of the stored channel numbers match the received channel number;

d. a default storage location for storing a default address value; and

e. an output circuit coupled to the comparing circuit, to the plurality of storage locations and to the default storage location to provide the stored address value within the storage location having the stored channel number matching the received channel number as an output address value, if one of the stored channel numbers within valid storage locations matches the received channel number, and to provide the default address value as the output address value if none of the stored channel numbers within the valid storage locations match the received channel number.

25. The receiving device as claimed in claim 24 wherein the comparing circuit only compares stored channel numbers within the valid storage locations to the received channel number.

26. The receiving device as claimed in claim 25 wherein the storage locations are programmable.

27. The receiving device as claimed in claim 26 wherein the host device sets the valid bit within the storage location when the storage location is programmed.

28. The receiving device as claimed in claim 27 further comprising a processing device coupled to the output circuit to receive the output address value, wherein the output address value specifies a beginning location for a series of instructions to be used to process the data received on the received channel number.

* * * * *